

#3  
2-2602  
ed

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re U.S. Patent Application of )  
KONDOH et al. )  
Application Number: To Be Assigned )  
Filed: Concurrently Herewith )  
For: PROGRAM COUNTER (PC) RELATIVE )  
ADDRESSING MODE WITH FAST DISPLACEMENT )

11017 U.S. PTO  
10/017198  
12/18/01

Honorable Assistant Commissioner  
for Patents  
Washington, D.C. 20231

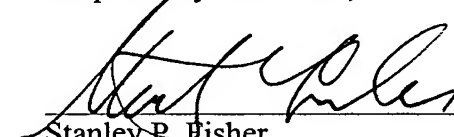
**REQUEST FOR PRIORITY  
UNDER 35 U.S.C. § 119  
AND THE INTERNATIONAL CONVENTION**

Sir:

In the matter of the above-captioned application for a United States patent, notice is hereby given that the Applicant claims the priority date of February 2, 2001, the filing date of Japanese patent application 2001-026253.

The certified copy of Japanese patent application 2001-026253 is being submitted herewith. Acknowledgment of receipt of the certified copy is respectfully requested in due course.

Respectfully submitted,

  
Stanley P. Fisher  
Registration Number 24,344

**REED SMITH HAZEL & THOMAS LLP**  
3110 Fairview Park Drive  
Suite 1400  
Falls Church, Virginia 22042  
(703) 641-4200

December 18, 2001

31 00-2207 65  
KH

日 本 国 特 許 庁  
JAPAN PATENT OFFICE



別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2001年 2月 2日

出 願 番 号

Application Number:

特願2001-026253

出 願 人

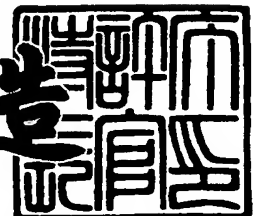
Applicant(s):

株式会社日立製作所

2001年10月19日

特許庁長官  
Commissioner,  
Japan Patent Office

及川耕造



出証番号 出証特2001-3092531

【書類名】 特許願

【整理番号】 H00022071A

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 9/38

【発明者】

【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目280番地 株式会社日立製作所中央研究所内

【氏名】 近藤 雄樹

【発明者】

【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目280番地 株式会社日立製作所中央研究所内

【氏名】 西井 修

【特許出願人】

【識別番号】 000005108

【氏名又は名称】 株式会社 日立製作所

【代理人】

【識別番号】 100075096

【弁理士】

【氏名又は名称】 作田 康夫

【電話番号】 03-3212-1111

【手数料の表示】

【予納台帳番号】 013088

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【ブルーフの要否】 要

【書類名】明細書

【発明の名称】高速ディスプレースメント付きPC相対分岐方式

【特許請求の範囲】

【請求項1】

命令語を一時的に格納するキャッシュもしくはバッファを具備し、命令実行時に命令語を該キャッシュもしくは該バッファから読み出すプロセッサであり、命令語がディスプレースメント付きPC相対アドレッシングを用いる命令である場合に実効アドレスの下位部分の計算を行い、その結果によって命令語中のディスプレースメントの値を置き換えて該キャッシュもしくは該バッファに格納する回路を有し、命令実行時の実効アドレス計算では該キャッシュもしくは該バッファに格納された実効アドレスの下位部分の結果を利用し、実効アドレスを高速に出力することを特徴とするプロセッサ。

【請求項2】

請求項1に記載したプロセッサであって、命令語を一時的に格納するキャッシュもしくはバッファ中に、命令語を格納する各エントリに1対1に対応する追加の記憶領域を有し、命令語がディスプレースメント付きPC相対アドレッシングを用いる命令である場合に行う実効アドレスの下位部分の計算結果によって情報を生成し、該記憶領域に該情報を格納することを特徴とするプロセッサ。

【請求項3】

請求項1に記載したプロセッサであって、実効アドレスが分岐先アドレスであるプロセッサ。

【請求項4】

請求項2に記載したプロセッサであって、実効アドレスが分岐先アドレスであるプロセッサ。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、計算機における中央処理装置に用いられるプロセッサに属する。特にプロセッサに用いられているディスプレースメント付きPC相対分岐命令に関す

る。

## 【0002】

### 【従来の技術】

従来から、プロセッサ内のレジスタの値と命令語に埋め込まれた定数（即値と呼ぶ）を実行時に加算し、その値を分岐先のアドレスとする分岐命令を備えたプロセッサが存在する。特にレジスタとして現在実行中の命令アドレスを示すプログラムカウンタ（PC）を用いた場合、分岐命令の位置からの相対位置を指定することができる。プログラムの配置アドレスに関わらず常に同じ位置へ分岐を行うことができるため、様々なプロセッサにおいて採用されている。ここで、分岐命令の命令語中の即値をディスプレースメント（以下 *disp* と略す）と呼び、前述の命令をディスプレースメント付き PC 相対分岐命令（以下  $PC + disp$  分岐命令と略す）と呼ぶ。例えばこれは「SH7750 プログラミングマニュアル」、日立製作所、1998 の pp. 10-22 に記載されている BRA 命令などが該当する。

$PC + disp$  分岐命令の実行には、（1）*disp* を多くの場合符号拡張し PC とビット幅をそろえ PC の値と符号拡張した *disp* を加算して分岐先アドレスを計算、（2）求まった値をアドレスとして命令をフェッチするという 2 段階の動作がある。符号拡張とはある幅  $m$  ビットを持った符号付き 2 進整数を幅  $n$  ( $> m$ ) ビットの符号付き 2 進整数に変換することを指し、変換元の整数を変換先の下位  $m$  ビットにコピーし、さらに変換元の最上位ビット、すなわち符号ビットを変換先の上位  $n - m$  ビットそれぞれにコピーすることにより実現できる。ここで（1）のアドレス計算の一例を図 1 に示す。

111 は実行する命令のアドレス、すなわち PC の値であり 32 ビットである。112 は  $PC + disp$  分岐命令の命令語であり、下位 12 ビットが *disp* である。命令語 112 のフォーマットを図 2 に示す。201 は命令オペコードであり、 $PC + disp$  分岐命令の場合命令オペコード 201 は 4 ビットである。202 および 203 は *disp* の一部であり、上位 1 ビットを 202、下位 11 ビットを 203 とした 204 は 12 ビット符号付きの *disp* の値となる。すなわち 202 は *disp* の符号ビットである。

分岐先アドレス 114 は、命令語 112 中の 204 を 32 ビット符号拡張した 113 と 111 を 32 ビット加算器 121 で加算することにより得られる。

PC+disp 分岐命令では分岐先アドレスを計算するために、本質的に  $n$  ビットの加算処理が必要である。 $n$  ビットの加算処理にはオーダ  $\log n$  の時間が必要となり、従来プロセッサでは本加算処理時間は不可避なものと考えられていた。

“Micro Processor Report,” September 13, 1999, pp.12-15, 22によると、上記加算処理時間を削減するためサンマイクロシステムズの MAJC アーキテクチャでは PC+disp 分岐命令の代わりにセミアブソリュート分岐（以下 Semi ABS 分岐と略す）命令を採用している。セミアブソリュート分岐命令ではアドレスのビットの一部分を絶対アドレスで指定する。図 3 に Semi ABS 分岐命令でのアドレス計算方法を示す。

311 は PC+disp 分岐命令と同様に PC の値である。312 は Semi ABS 分岐命令の命令語であり、図 4 に Semi ABS 分岐命令の命令フォーマットを示す。401 は命令オペコード及び分岐先アドレス計算に用いるオペランド以外のオペランド部分である。402 及び 403 が分岐先アドレス計算に用いるオペランド部分であり、命令語 311 の下位 12 ビット部分が 403、その上位 2 ビットが 402 となっている。

分岐先アドレス 314 の計算については、下位 2 ビットは“00”（定数）とし、その上位の 12 ビット部分は命令語 312 中の 403 がそのまま入る。上位 18 ビット部分については、命令語 312 中の 402 を 18 ビット符号拡張した 313 と 311 を 18 ビット加算器 321 で加算し、出力とする。

Semi ABS 分岐では PC+disp 分岐と比較して分岐先アドレス計算に必要な加算器の桁数が少なくて済むため、遅延時間を短く押さえることができ、動作周波数の向上が期待できる。

特に命令キャッシュを具備するプロセッサの場合、Semi ABS 分岐では高速化が可能となる。分岐命令の実行においては分岐先アドレス計算の次に命令キャッシュへのアクセスが必要となるが、通常キャッシュの動作は（1）アドレスのうちインデックス部となる部分をキャッシュアレイに入力し対応するタグ部とデ

ータを読み出し、(2) アドレスのタグ部と対応する部分と(1)で出力されたタグ部を比較し、一致する場合キャッシュヒットとしてデータを出力するという2段階に分かれている。すなわちキャッシュへのインデックス部入力、タグ部の入力よりもタイミング上先に行う必要がある。また、アドレスのうちインデックス部となる部分は、通常はアドレスの下位部分に配置されている。

Semi ABS分岐では分岐先アドレスの下位部分は加算を行うことなくそのまま出力されるため、インデックス部のビット範囲が加算されないビット範囲に含まれる時、分岐先アドレスの上位部分の計算の完了を待たずに並列にキャッシュアレイのアクセスを開始することができ、分岐命令の実行が全体として高速化される。

#### 【0003】

##### 【発明が解決しようとする課題】

上記の従来技術では高速な分岐を実現するために、PC+disp分岐の代わりに新しいSemi ABS分岐を導入している。しかし、Semi ABS分岐論理方式を採用する新たなプロセッサではSemi ABS分岐を採用していない命令によって書かれたプログラムを実行できない、すなわち、既存のPC+disp分岐命令を有するプロセッサ上で動作するオブジェクトコードにはSemi ABS分岐命令が含まれていないため、高速化の恩恵を受けることができない。

また、Semi ABS分岐では分岐先アドレスの一部分を絶対アドレスで指定する必要があるため、コンパイル時に分岐先アドレスの範囲に収まっているかどうか判定できない、コンパイル後のオブジェクトコードがリロケータブルにならない、という弊害がある。リロケータブルとは、オブジェクトコードの配置アドレスに関わらず、そのオブジェクトコードが実行可能であることを指す。

本発明が解決しようとする課題の1つは、以上の問題に鑑みプロセッサの命令セットを新規に用意、または既存の命令セットに命令を追加することなく、PC+disp分岐命令の動作を高速化することである。また、PC+disp分岐命令を削除する必要がないため、上記に示した弊害についても発生しない。

#### 【0004】

##### 【課題を解決するための手段】

本発明は命令語をキャッシュもしくはバッファに一時的に格納し、命令の実行時はキャッシュもしくはバッファに一時的に格納されている命令語を読み出して実行するプロセッサに適用するものである。

## 【0005】

本発明では、命令語をキャッシュもしくはバッファに格納する時に  $PC + disp$  アドレッシング命令であるかどうかを判定し、 $PC + disp$  アドレッシング命令である場合にはキャッシュもしくはバッファへの格納前にあらかじめ  $PC + disp$  の計算の下位部分を行い、命令語の  $disp$  部分とキャッシュもしくはバッファに追加されたビットに計算結果を書き込む。該  $PC + disp$  アドレッシング命令をキャッシュもしくはバッファから読み出して実行する時には  $PC + disp$  計算の下位部分はあらかじめ計算された結果をそのまま出力し、下位アドレス部分の計算を省略することにより、 $PC + disp$  分岐命令のアドレス計算に要する時間を削減する。以上が本発明の手段である。

## 【0006】

## 【発明の実施の形態】

以下に、本発明の実施例を2例示す。

## （実施例1）

図5及び図6は本発明を最も単純な形で実現した例である。図5は命令フェッチ部、すなわち主記憶から読み出した命令語を命令キャッシュに格納する部分である。511は主記憶から読み出した命令語であり、命令フォーマットは図2の112と同等である。本実施例のプロセッサではアドレスは32ビットである。512は命令が格納されている主記憶上のアドレスであり、幅は32ビットである。また、アドレス512は命令語511の実行時のPCの値に等しい。

## 【0007】

501は命令キャッシュである。514は命令語511と同じ幅をもっており、命令キャッシュ501内に複数のエントリが存在する。515は514と1対1で対応しており、幅は1ビットである。回路ブロック502は命令語511とアドレス512を入力とし、出力が命令キャッシュ501内の514及び515に格納される。



## 【0008】

回路ブロック502の動作について説明する。命令語511内の201および202は命令キャッシュ501内の514の対応する部分にそのまま格納される。522はデコーダであり、命令語511内の201がPC+disp分岐命令かどうかを判定し、セクタ523の出力を切り替える。命令語511内の201がPC+disp分岐命令以外の場合、セクタ523は命令語511内の203をそのまま出力し、命令キャッシュ501内の514の203に対応する部分にそのまま格納される。

命令語511内の201がPC+disp分岐命令を示している場合、命令語511内の203とアドレス512の下位11ビット部分を加算器512で加算し、得られた11ビットの結果をセクタ523が出力する。命令キャッシュ501内の514の203に対応する部分にはこの結果が格納される。また、加算器521の上位へのキャリービット出力は515に格納される。

図6はPC+disp分岐命令実行時の分岐先アドレス計算部である。611はPCの値であり、命令キャッシュ501内に格納された514と515及び611の値を用いて、分岐先アドレス613を求める。514内の203に対応する部分はそのまま分岐先アドレス613の下位11ビット部分に出力される。612は514内の202に対応する部分を21ビットに符号拡張したものである。加算器621は612と611の上位21ビット部分とを加算し、結果を分岐先アドレス613の上位21ビット部分に出力する。

## 【0009】

図5に示した回路ブロック502では、dispの下位11ビット部分とPCの下位11ビット部分の加算を行い、515に下位からのキャリービット出力を保存している。また、図6に示したPC+disp分岐命令の実行時にはdispの上位1ビットを21ビット符号拡張したものとPCの上位21ビット部分を、515に保存された下位からのキャリービットを考慮して加算している。すなわち結果として算出される分岐先アドレス613は、12ビットのdispを32ビットに符号拡張してPCと加算した値と等しい。

(実施例2)

図7、図8は本発明の別の実施例である。図7は実施例1の図5と同じく命令フェッチ部である。701, 711, 712および714は実施例1での501, 511, 512および514に相当し、回路ブロック702は実施例1での回路ブロック502に対応する。実施例1での515に相当する部分はこの実施例には存在しない。

デコーダ722とセレクタ723の動作も実施例1でのデコーダ522とセレクタ523の動作とほぼ同一であり、命令語711内のオペコード501がPC+disp分岐命令である場合に加算器721の出力である713を命令キャッシュ701内の714の204に対応する部分に出力し、それ以外の場合は命令語711内の204をそのまま714内の204に対応する部分に出力する。加算器721は命令語711内の204とアドレス712の下位12ビットを加算し、12ビットの加算結果を713に出力する。

#### 【0010】

図8は実施例1の図6と同じくPC+disp分岐命令実行時の分岐先アドレス計算部である。811はPCであり、命令キャッシュ701内に格納された714及び811の値を用いて、分岐先アドレス812を求める。714内の204に対応する部分は、そのまま分岐先アドレス812の下位12ビット部分に出力される。

824及び825は加算器であり、それぞれ811の上位20ビットの値に+1, -1を加算した値を出力する。すなわちセレクタ823は、811の上位20ビットの値について、+1, ±0, -1した値のどれかを、デコーダ822の出力に従って分岐先アドレス812の上位20ビット部分に出力する。

#### 【0011】

デコーダ822には、(a) 714内の203に対応する部分と811の下位11ビットを比較器821で比較した結果(203に対応する部分の方が大きい、もしくは等しい場合を0、逆の場合を1とする)、(b) 714内の502に対応する部分、(c) 811の下位から12ビット目が入力される。デコーダ822の入力に対するセレクタ823の出力の対応を図10に示す。

比較器821の出力は、命令フェッチ部で行った11ビット加算の、下位10ビ

ット目から11ビット目へのキャリービットである。このキャリービットと714内の202に対応する部分と811の下位から12ビット目の排他的論理和を取ると、命令語711内の202、すなわち204の符号ビットを算出することができる。

#### 【0012】

また、前述のキャリービットと202、811の下位から12ビット目の値について、1ビット値3個の加算を行いキャリービットを求めると、命令フェッチ部で行った加算の、最上位ビットからのキャリービットが求められる。

202を20ビット符号拡張し、811の上位20ビット部分と、命令フェッチ部で行った加算の最上位ビットからのキャリービットを加算すれば、分岐先アドレス812は12ビットのdispを32ビットに符号拡張してPCと加算した値と等しくなる。しかし、(1)202が1の場合に811の上位20ビットに-1を加算し、加えて(2)命令フェッチ部で行った加算の最上位ビットからのキャリービットが1の場合に+1を加算すれば、先ほど述べた加算と同じ動作ができる。したがって、811の上位20ビットに対して、202の値と命令フェッチ部で行った加算の最上位ビットからのキャリービットが、00もしくは11の場合はそのまま、01の場合は+1を加算、10の場合は-1を加算すれば、先ほど述べた加算と同じ結果が得られる。

#### 【0013】

以上実施例を説明したが、本発明はその技術的思想のもとに種々の変形が可能である。例を示すと、(1)分岐命令のみでなく、PC+dispアドレッシングによるロードストア命令に実施可能である、(2)dispの幅が一種類のみでなく8、12のように複数種類あっても実施例1、2とも実施できる、といった変形が可能である。

#### 【0014】

##### 【発明の効果】

本発明の実施例1に示したプロセッサの第一の利点は、SemiABS分岐命令と同様、PC+disp分岐命令の実行時の分岐先アドレス計算を高速に処理、すなわち分岐先アドレス計算にかかる遅延時間を短縮できることである。これに

より特にパイプラインによって命令を実行するプロセッサにおいては、分岐先アドレスの計算を行うステージに必要となる時間を減少させることができ、プロセッサの動作周波数向上、あるいは処理サイクル数の削減、あるいはその両方が可能となる。

例えば、実施例 1 では全アドレスが 3 2 ビット、`disp` が 1 2 ビットであり従来の `PC+disp` 分岐論理方式の場合、`PC+disp` 分岐命令の実行時には 3 2 ビット加算が必要となるが、実施例 1 での `PC+disp` 分岐論理方式の場合では 2 1 ビット加算で済む。全アドレスが 6 4 ビット、`disp` が 2 0 ビットであるとした場合では、従来の `PC+disp` 論理方式では 6 4 ビット加算、実施例 1 では 4 5 ビット加算で済む。

また、実施例 1 に示したプロセッサにおいては、`SemiABS` 分岐命令と同様に命令キャッシュへのアクセスを早い時刻に開始することによって `PC+disp` 分岐命令の動作を全体的に高速化できるという第二の利点がある。

実施例 1 に示したプロセッサでの第三の利点は、ある `PC+disp` 分岐命令が命令キャッシュにヒットして繰り返し実行される場合には、分岐先アドレスの下位部分の計算は初回実行時、すなわち命令キャッシュに格納される時にのみ行われ、以降の実行時に重複して計算されることを回避できる点である。図 9 に従来の `PC+disp` 分岐論理方式の場合と実施例 1 での `PC+disp` 分岐論理方式の場合での、タイミングチャートでの比較を示す。

9 0 1 は従来の `PC+disp` 分岐論理方式の場合であり、9 1 1 は命令キャッシュ読み出し、および命令キャッシュミスヒットによる主記憶からの命令語読み出し動作、9 1 3 は 9 1 1 で読み出した命令語をキャッシュに格納する動作、9 1 4 は分岐先アドレスの下位部分の加算動作、9 1 5 は分岐先アドレスの上位部分の加算と分岐先アドレス命令の命令キャッシュからの読み出し動作である。また、9 1 2 は命令キャッシュヒットの場合の読み出し動作である。9 0 2 は実施例 1 での `PC+disp` 分岐論理方式の場合であり、ある `PC+disp` 分岐命令が命令キャッシュにヒットする状態で繰り返し実行される場合、繰り返し回数が多いほど実施例 1 での `PC+disp` 分岐論理方式の方が、実行に要する時間が短縮される。

【0015】

実施例1に示したプロセッサの第四の利点は、第一、第二及び第三の利点として挙げた効果を、既存のPC+disp型アドレッシングを備えるプロセッサの命令セットに変更を加えることなく、享受できる点である。

本発明の実施例2に示したプロセッサでは、実行時にdispの符号ビットおよび命令フェッチ時の加算の最上位ビットからのキャリービットの再計算を行うため実施例1に示したプロセッサの第一の利点に相当する効果は実質的にはほとんど得られないが、第二、第三および第四の利点に関しては同等に効果が得られる。

また、実施例2に示したプロセッサでは、主記憶から読み出した命令語を格納するキャッシュもしくはバッファに追加の記憶領域が不要であるという利点がある。

【図面の簡単な説明】

【図1】

従来のPC+disp分岐での分岐先アドレス計算法。

【図2】

命令語112のフォーマット。

【図3】

従来のSemiABS分岐での分岐先アドレス計算法。

【図4】

命令語312のフォーマット。

【図5】

本発明の第一の実施例における命令語の命令キャッシュ格納時の処理回路。

【図6】

本発明の第一の実施例におけるPC+disp分岐命令実行時のアドレス計算回路。

【図7】

本発明の第二の実施例における命令語の命令キャッシュ格納時の処理回路。

【図8】

本発明の第二の実施例におけるPC+disp分岐命令実行時のアドレス計算回路。

【図9】

従来のPC+disp分岐論理方式の場合と実施例1でのPC+disp分岐論理方式の場合のタイミングチャートでの比較。

【図10】

デコーダ822への入力に対するセレクタ823の出力対応図。

【符号の説明】

111-PCの値, 112-PC+disp分岐命令の命令語, 114-PC+disp命令での分岐先アドレス, 121-32ビット加算器。

311-PCの値, 312-SemiABS分岐命令の命令語, 314-SemiABS分岐命令での分岐先アドレス, 321-18ビット加算器。

501-命令キャッシュ, 511-PC+disp分岐命令の命令語, 512-読み出した命令の主記憶上のアドレス値, 514-命令キャッシュ内のエントリ, 515-514に一对一に対応する記憶領域, 521-11ビット加算器, 522-デコーダ, 523-セレクタ。

611-PCの値, 613-分岐先アドレス, 621-21ビット加算器。

701-命令キャッシュ, 711-PC+disp分岐命令の命令語, 712-読み出した命令の主記憶上のアドレス値, 714-命令キャッシュ内のエントリ, 721-12ビット加算器, 722-デコーダ, 723-セレクタ。

811-PCの値, 812-分岐先アドレス, 821-11ビット比較器, 822-デコーダ, 823-セレクタ, 824, 825-20ビット加算器。

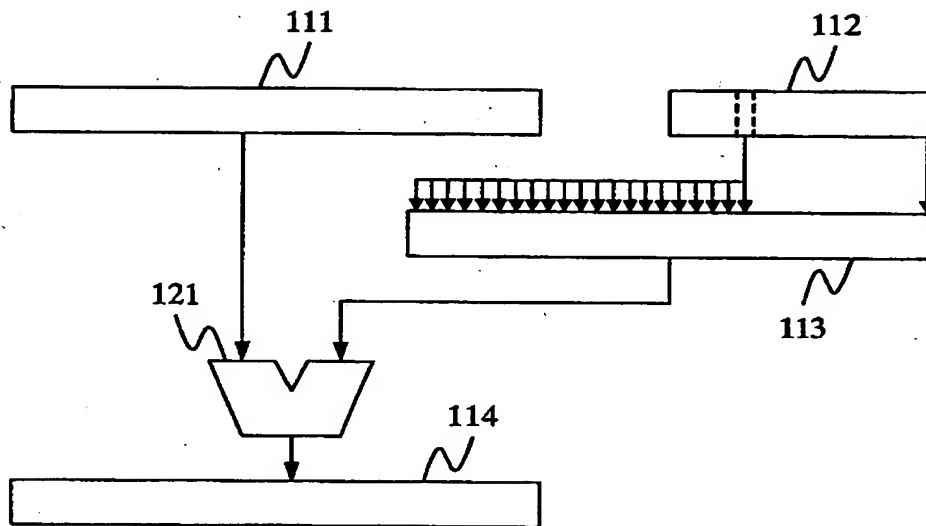
901-従来のPC+disp分岐論理方式の場合のPC+disp分岐命令動作, 902-本発明の第一の実施例でのPC+disp分岐論理方式の場合のPC+disp分岐命令動作, 911-命令キャッシュ読み出し及び命令キャッシュミスヒットによる主記憶からの命令語読み出し動作, 912-命令キャッシュヒット時の命令語読み出し動作, 913-911で読み出した命令語をキャッシュに格納する動作, 914-分岐先アドレスの下位部分の加算動作, 915-分岐先アドレスの上位部分の加算と分岐先アドレス命令の命令キャッシュからの読

み出し動作。

【書類名】 図面

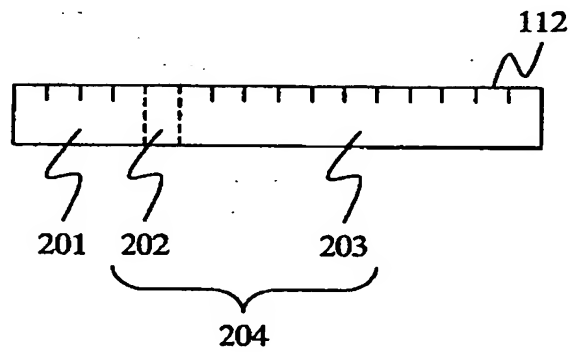
【図1】

図 1



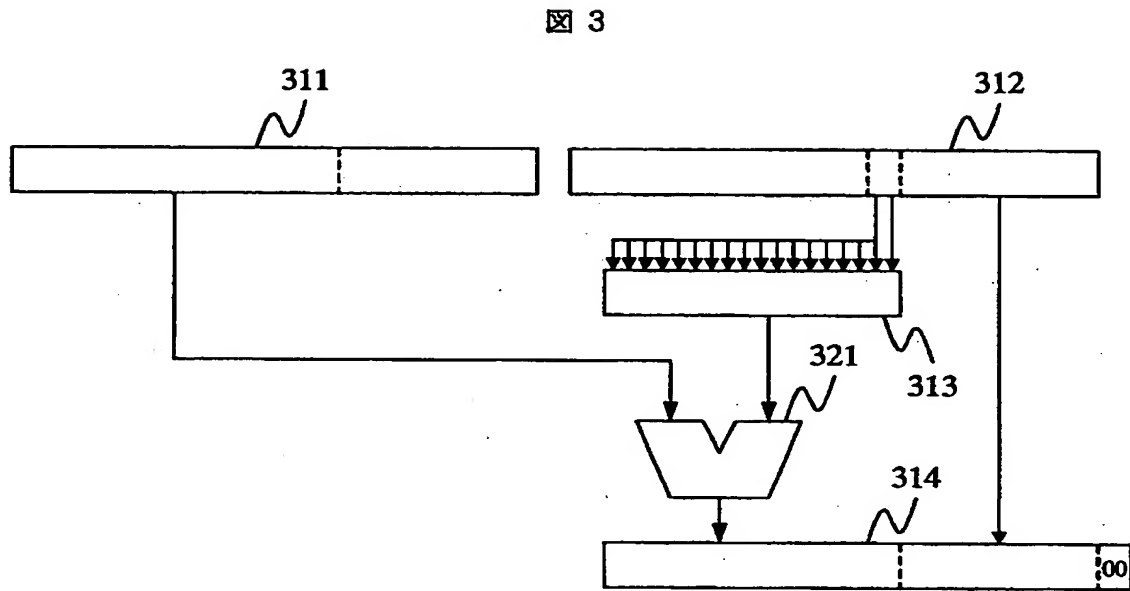
【図2】

図 2

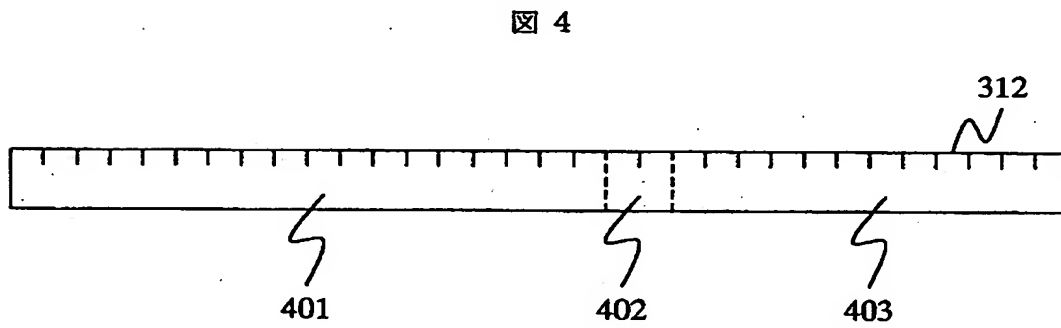




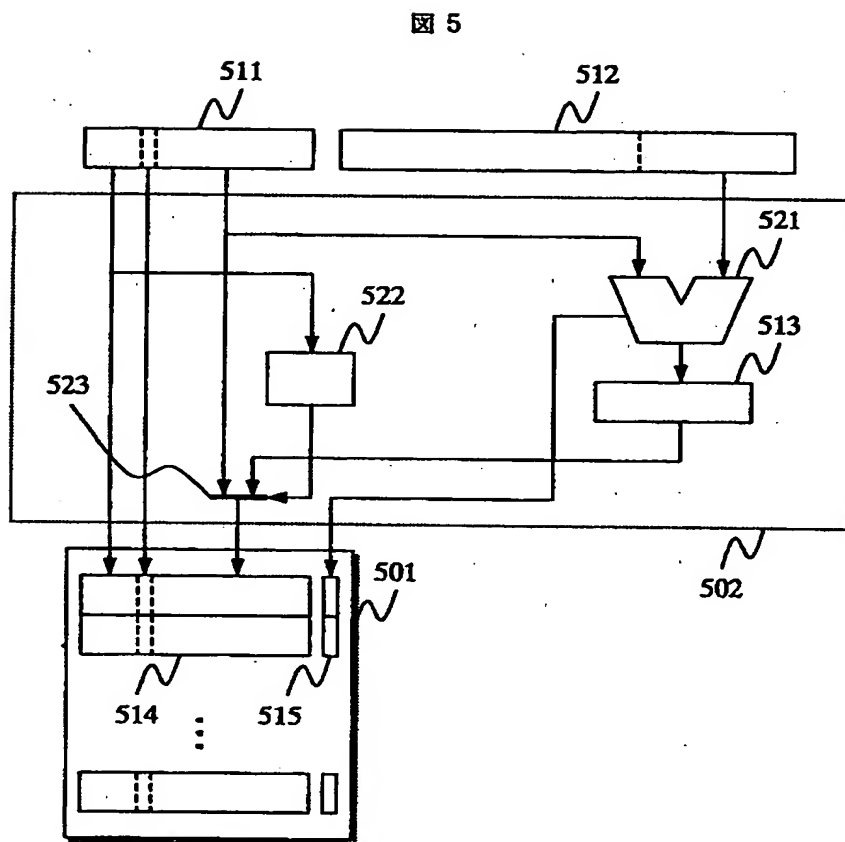
【図 3】



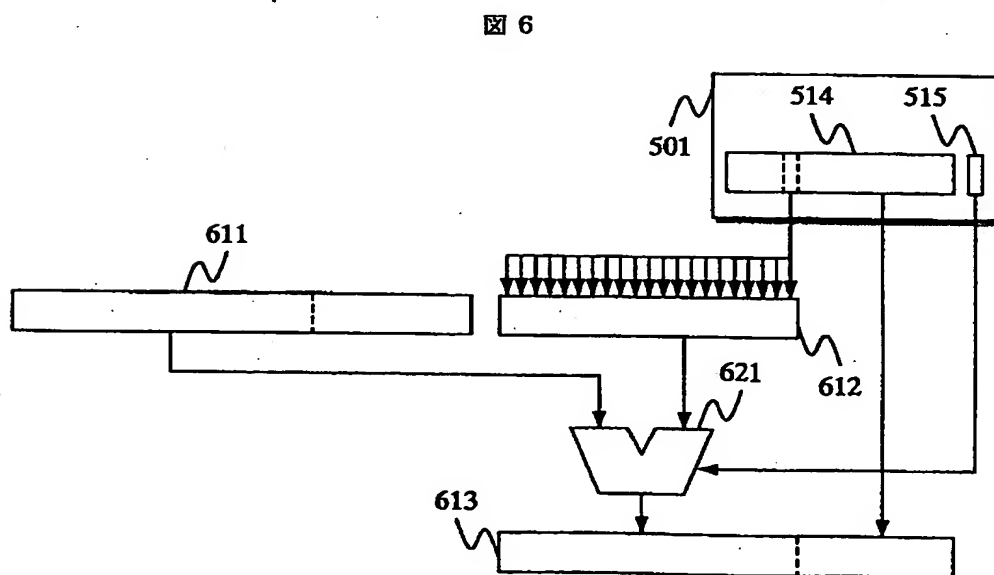
【図 4】



【図 5】

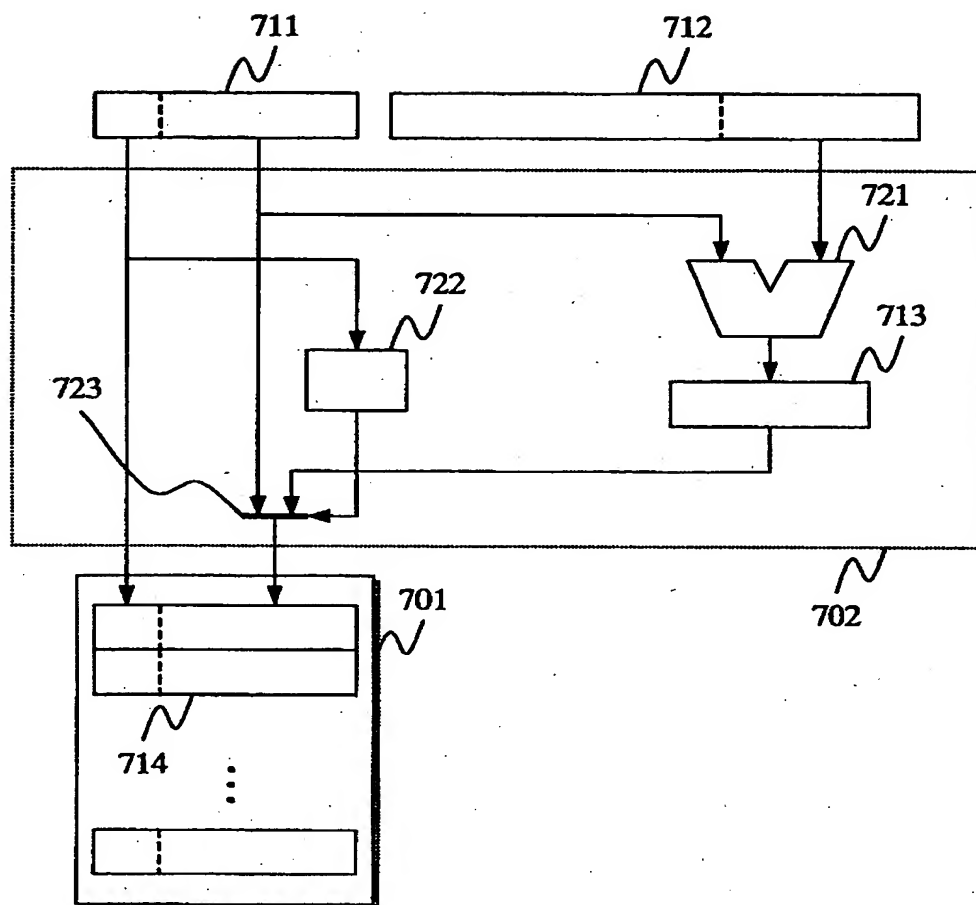


【図 6】



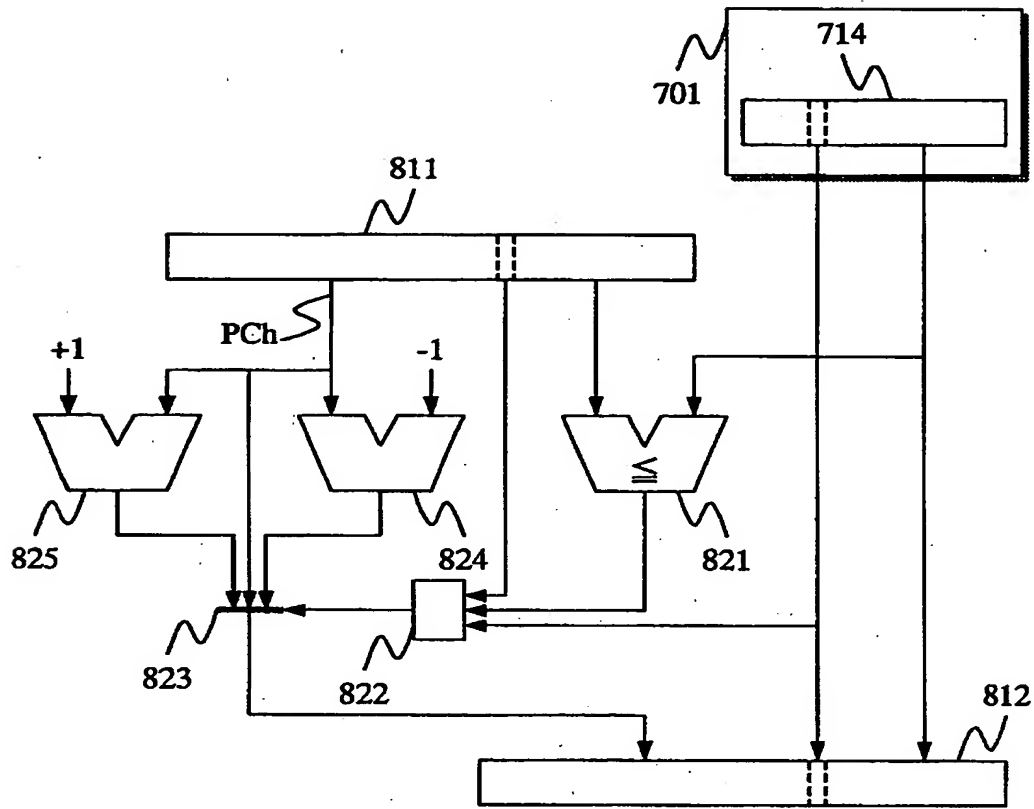
【図7】

図7



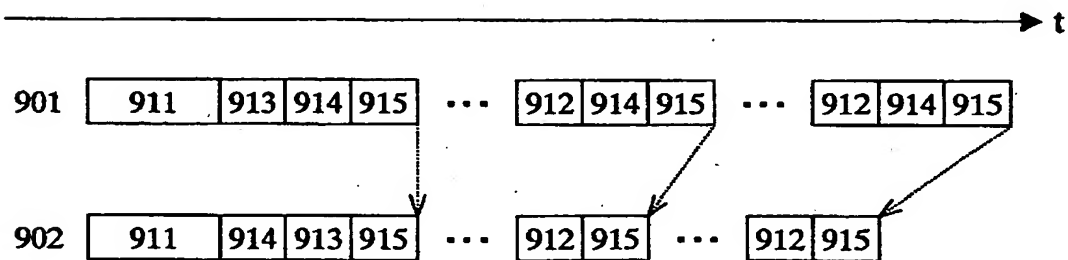
【図 8】

図 8



【図 9】

図 9



【図 1 0】

図 1 0

入力	(a)	1	1	1	1	0	0	0	0
	(b)	1	1	0	0	1	1	0	0
	(c)	1	0	1	0	1	0	1	0
出力	セクタ823 の 出力	PCh を 選択	PCh を 選択	PCh+1 を 選択	PCh を 選択	PCh を 選択	PCh-1 を 選択	PCh を 選択	PCh を 選択

【書類名】 要約書

【要約】

【課題】 既存のプロセッサの命令動作を変更，もしくは新規命令を追加することなく，ディスプレースメント付きPC相対分岐命令の動作を高速化する。

【解決手段】 命令語をキャッシュもしくはバッファに格納する前にあらかじめ分岐先アドレス計算の下位部分を行い，命令語のディスプレースメント部分及び該キャッシュもしくは該バッファに追加されたビットに計算結果を書き込み，命令実行時は該キャッシュもしくは該バッファに格納された計算結果を利用して分岐先アドレスの下位部分の計算を省略することで分岐先アドレス計算を高速化する。

【選択図】 図6

特2001-026253

認定・付加情報

特許出願の番号	特願2001-026253
受付番号	50100145991
書類名	特許願
担当官	第七担当上席 0096
作成日	平成13年 2月 5日

<認定情報・付加情報>

【提出日】	平成13年 2月 2日
-------	-------------

次頁無

出 願 人 履 歴 情 報

識別番号 [000005108]

1. 変更年月日 1990年 8月31日

[変更理由] 新規登録

住 所 東京都千代田区神田駿河台4丁目6番地  
氏 名 株式会社日立製作所